

CS



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/559,531	04/28/2000	Georgios Chrysanthakopoulos	03797.84196	3281

28319 7590 11/09/2004

BANNER & WITCOFF LTD.,
ATTORNEYS FOR MICROSOFT
1001 G STREET, N.W.
ELEVENTH STREET
WASHINGTON, DC 20001-4597

EXAMINER

CRAIG, DWIN M

ART UNIT	PAPER NUMBER
----------	--------------

2123

DATE MAILED: 11/09/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/559,531

Applicant(s)

CHRYSTHAKOPOULOS,
GEORGIOS

Examiner

Dwin M Craig

Art Unit

2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 August 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-39 is/are pending in the application.
- 4a) Of the above claim(s) 12-16 is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-11 and 17-36 is/are rejected.
- 7) ☒ Claim(s) 37-39 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

1. **Claims 1-11 and 17-36** have been presented for reconsideration in view of Applicant's amendments and arguments. **Claims 12-16** have been cancelled. Dependent **Claims 37-39** have been presented for Examination.

Response to Arguments

2. The arguments set forth in Applicant's response dated 8-13-2004 have been fully considered. The Examiner's response is as follows.

2.1 The Examiner thanks the Applicant's for submitting the corrected drawing and hereby withdraws the objections to the drawings.

2.2 As regards the Applicant's response to the 35 U.S.C. 103(a) rejections of Claims 1, 17 and 33.

Applicant argued:

(From page 11 of 17, 8-13-2004 response)...

Regarding independent claim 1, the action alleges that Bonola, at col. 1, line 65 to col. 2, line 32, discloses a method of emulating a device including loading an emulation driver for the device and dynamically exposing the emulated device functionality. The action acknowledges however that Bonola fails to teach or suggest creating a virtual device object for the device and emulating a device by a node on a serial bus. To show creating a virtual device object for the device, the action relies on Tushie at col. 3, lines 19-24 and col. 4, lines 45-55. To show emulating a device by a node on a serial bus, the action relies on Staats claiming it discloses device drivers providing support for devices on nodes of a serial bus pointing to figures 1, 4, 7 and col. 2, lines 44-52.

The Examiner respectfully traverses the Applicant's argument. The Examiner argues that given the disclosed functionality of the IEEE 1394 bus standard that an artisan would be motivated to provide for virtual device drivers to represent the different devices, of which there

Art Unit: 2123

is a large variety, to be accommodated on the IEEE 1394 bus standard. For example and as disclosed in the *Staats* reference in **(Col. 3 Lines 39-48)** where there is disclosed, *"The bus architecture described herein, though described with reference to a preferred embodiment comprising components for a single computer, in general has a broader scope and could include test and measurement systems, positioning and robotic systems, and audio and video components for example. ... The present invention may be applied to any arbitrary assembled collection of nodes linked together as a network of devices."*

The Examiner notes that for different devices there are already large amounts of software required to support the different device families. For example, NDIS drivers required a network protocol stack, File system drivers require a stack of file system filter drivers, when developing a new product, such as disk drive that has a 1394 interface, an artisan of ordinary skill could write a virtual driver for the 1394 bus to support the file system drivers further up the stack and also provide for *link layer* and *transaction layer* support of the 1394 serial bus. The hardware specific aspects of the driver in regards to the final product, *i.e. the disk drive with a 1394 interface*, could be stubbed out for future coding. The artisan would now be able to debug the portions of the device driver dealing with interfacing with the file system driver stack and 1394 bus protocol stack without having to wait for the first prototype disk drive to be provided **(Bonola, Col. 1 Lines 43-57)** *"In many instances, the actual peripheral device hardware may not be available while the device driver is being developed by the manufacturer. As a result, actual testing and any debugging changes that need to be made must wait until the actual hardware becomes available. This increases the development time for the device driver and as a result, the peripheral device, thereby delaying the availability of the new peripheral device. If a large*

Art Unit: 2123

portion of the device driver testing and debugging could occur without the need for the actual peripheral device hardware, the overall time to develop a peripheral device would be decreased, resulting in more rapid computer system improvement. It is further desirable that this testing be performed under conditions close to those present if the peripheral device were present." The Examiner stands by the earlier motivations to combine the references and has used the preceding discussion to clarify the rational in combining the different references used in the 35 U.S.C. 103(a) rejections of the last office action.

2.3 Regarding the Applicant's response to the 35 U.S.C. 103(a) rejections of Claim 19, applicant argued:

Indeed, it would appear that the action engaged in an exercise of impermissible hindsight. Namely, with the claim 19 invention in hand and the references, the action tried to create a reason to combine Staats and Bonola. In any event, applicants submit that one skilled in the art would not have had motivation to combine Staats and Bonola as asserted in the action to obtain the claim 19 inventions.

The Examiner respectfully traverses the Applicant's arguments. The Examiner notes the following, Virtual Device Drivers that emulate peripherals are known in the art, *Bonola*, Serial busses are known in the art, the *IEEE 1394* standard is known in the art, *Staats*, and therefore in order to reduce to practice a 1394 bus an artisan would have to accommodate the *development* of device drivers on the serial bus in order to be able to bring to market 1394 devices in an efficient manner. *(Please see section 2.2 above.)*

2.4 Regarding the Applicant's response to the rejection of independent Claim 17.

Applicant argued;

Art Unit: 2123

(From page 12 of 17, 8-13-2004 response)...

It follow(s) that Bonola lacks a teaching or suggestion of allocating node address space to intercept requests to an emulated device register. Indeed, the action does not point with specificity to any portion of Bonola, Tushie or Staats for these claim 17 features.

The Examiner will now map the specific limitations to the references. *Specifically*, the Applicant's claims are directed towards the allocation of memory for the purpose of *emulation* of a specific hardware device and more specifically emulation of a particular device's control and data registers. The *Bonola* reference discloses allocation of memory for the purpose of device emulation (***Bonola*, Figure 3A Item 212 "ALLOCATE CONTINUOUS MEMORY FOR EMULATOR", Col. 8 Lines 31-47**) and providing for emulation of the virtual emulated devices control and data registers in memory (**Col. 9 Lines 67, Col. 10 Lines 1-5**). As regards the limitation of allocating node address space the *Staats* reference discloses the, *known in the USB art*, method of loading device drivers and allocating, *node address spaces*, in computer systems using a USB serial bus (***Staats*, Col. 2 Lines 44-53, Col. 6 Lines 65-67, Col. 7 Lines 1-13**). The Examiner notes that there are addresses provided in the CSR memory of a device on the USB serial bus and that it would be *inherent*, in the *Staats* reference for there to be memory space allocated for these addresses in order to load the correct device driver for emulation as disclosed in the *Bonola* reference.

2.5 As regards the Applicant's response to the motivation to combine the *Bonola* and *Tushie* references.

Applicant argued;

(From page 12 of 17, 8-13-2004 response)...

Art Unit: 2123

Contrary to the action's position, Applicant respectfully submits that one skilled in the art would not have been motivated to combine Bonola and Tushie. The action contends that one would have used the virtual device object described in Tushie with the emulation technique of Bonola so that a configuration change can take place on the device model rather than the device itself. The problem with this position is that one of the primary reasons that Bonola emulates the peripheral device is to allow for device drivers to be tested without the device itself. Indeed the emulation software of Bonola could be easily modified without having to use a complex programming language which would add complexity and expense, and without the need to utilize virtual device objects of Tushie as proposed in the action. In short the motivation for combining the Bonola and Tushie proposed in the action is illusory. Staats fails to mitigate this defect with the propriety of the combination. For at least this reason, one would not have combined Bonola, Tushie and Staats to obtain the invention of independent claims 1, 17 and 33.

The Examiner respectfully traverses Applicant's argument. The Examiner notes that the *Staats* reference is disclosing a method of handling different types of peripherals on a serial USB bus. The methods of using a virtual emulation of a possible future USB device for the purpose of testing a driver before the actual hardware device is ready would be very useful to an artisan for the same reasons for usefulness put forward in the *Bonola* reference namely that the reduction in development time (*Bonola Col. 1 Lines 40-64*), more than justifies the level of effort required to develop a device driver that emulates a future peripheral. Further, the need to test and confirm that the future device driver can properly function correctly in a stack of drivers and properly support different communications protocols, as is required in the Microsoft® Windows® operating system, *file system filter driver stacks and NDIS driver stacks come to mind*, more than justifies using the emulation methods disclosed in the *Bonola* reference with the USB serial bus methods disclosed in the *Staats* reference.

2.6 Regarding the Applicants response to the 35 U.S.C. 103 rejection of Claim 19.

Applicant argued:

(From page 13 of 17, 8-13-2004 response)...

Art Unit: 2123

Independent claim 19 is directed to a system for emulating a device and calls for a serial bus and a node connected to the serial bus, which is configured to emulate at least one device. The action contends and applicant agrees that Staats discloses all the claim 19 features, but for the node emulating a device. To overcome this deficiency, the action relies on Bonola. Bonola describes emulating a peripheral device to allow device driver development before availability of the peripheral device.

And...

(From page 14 of 17, 8-13-2004 response)...

Thus, the action has identified an alleged motivation for combining the references based on the common knowledge of one of ordinary skill in the art. Applicant respectfully disagrees with the action's analysis that one skilled in the art would have motivated to modify Staats with Bonola as asserted in the action.

The Examiner respectfully traverses Applicant's arguments. The *Bonola* reference discloses that emulation of a peripheral is known in the art, and clearly with the advent of newer technologies, such as USB serial interfaces, an artisan of ordinary skill would be motivated to use "*tried and true*" methodologies, as disclosed in the *Bonola* reference, for the same reasons argued in paragraph 2.5 of this Office Action.

Applicant further argued *(From page 15 of 17, 8-13-2004 response)...*

In sum, applicant believes that Staats provides sufficient information to teach away from the motivation identified in the action. Indeed, it would appear that the action engaged in an exercise of impermissible hindsight. Namely, with the claim 19 invention in hand and the references, the action tried to create a reason to combine Staats and Bonola. In any event, applicants submit that one skilled in the art would not have had motivation to combine Staats and Bonola as asserted in the action to obtain the claim 19 invention.

The Examiner respectfully traverses the Applicant's arguments. The Applicant's arguments are directed towards a supposed deficiency of the references cited in that the Applicant believes that the *Staats* reference actually teaches away from the limitations disclosed in the *Bonola* reference. The Examiner notes that the development of device drivers can be a

Art Unit: 2123

long and complicated task, which requires significant expense and time for highly skilled artisans to perform. The *Bonola* reference provides a method for performing the expensive and time-consuming task of device driver development to proceed without the need for finished and debugged hardware to be present. Further, the methods and technologies disclosed in the *Staats* reference provide for a flexible method of loading and enumerating device drivers such that the methods of device emulation disclosed in the *Bonola* reference would be facilitated. The Examiner asserts that the earlier motivations to combine where proper.

2.7 Regarding the Applicants arguments regarding newly submitted dependent Claim 37.

Applicant argued, (*From page 16 of 17, 8-13-2004 response*)...

In contrast to the claim 37 invention however, Tushie describes in the cited portion set for the above that the virtual device objects are configured to reflect the physical personalization equipment present in the production environment. Tushie further states that when the production environment changes, the user simply changes the configuration of the corresponding virtual device objects.' Col. 3, lines 16-18. It is quite clear that Tushie neither teaches nor suggests creating a virtual device object for the device without the device being connected as recited in claim 37.

The Examiner agrees with the Applicant's argument in regards to newly submitted Claim 37 in that the *Tushie* reference fails to teach or suggest the claimed limitation of dependent Claim 37.

2.8 An updated search has revealed new art.

Claim Interpretation

3. The Applicant's claim language has been given the broadest reasonable interpretation. The Examiner has interpreted the term "*virtual device object*" to mean, any piece of software or computer executable code that, while residing in a computer memory "*abstracts*" a hardware

Art Unit: 2123

device and provides various “*entry points*” or software functions, in the “*known in the art*” methods of providing a “*handle*” in order to send and receive data and control functions/commands to the hardware device, from other software applications or “*device drivers*” in either of one of the following two cases;

1. An actual piece of hardware is present in the computer system, or
2. A piece of computer hardware is being abstracted and might not actually be present in the computer system at the time the “*virtual device object*” or “*virtual device driver*” is being accessed/interacted with.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Independent **Claim 1** is rejected under 35 U.S.C. 102 (b) as being anticipated by **Goff et al. U.S. Patent 5,835,791**.

4.1 As regards independent **Claim 1** the *Goff et al.* reference teaches, emulating a device by a node on a serial bus (**Figure 3, items 87, 88 & 89 and Col. 5 Lines 60-63, Col. 6 Lines 5-7**), creating a virtual device object for the device (**Figure 5 and Col. 3 Lines 40-56**), *the Examiner asserts that creating a virtual serial keyboard block, as disclosed in Figure 5 Item 129 is functionally equivalent to creating a virtual device object, responsive to the step of creating*

Art Unit: 2123

the virtual device object, loading an emulation driver for the device (**Col. 2 Lines 57-65**), and dynamically exposing, on the serial bus an emulated device functionality (**Figure 1, Col. 2 Lines 42-65**).

4.2 As regards dependent **Claim 6** the *Goff et al.* reference teaches a device being plugged into a serial bus (**Figure 1 Item 16**).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
 2. Ascertaining the differences between the prior art and the claims at issue.
 3. Resolving the level of ordinary skill in the pertinent art.
 4. Considering objective evidence present in the application indicating obviousness or nonobviousness.
- 5. Independent Claims 1, 17 and 33 and dependent Claims 2, 3, 6-11, 18 and 34-36** rejected under **35 U.S.C. 103(a)** as being unpatentable over **Bonola U.S. Patent 5,717,903** in view of **Tushie et al. U.S. Patent 6,202,155** and in further view of **Staats U.S. Patent 5,968,152** and in further view of "*OFFICIAL NOTICE*".

5.1 As regards independent **Claims 1, 17 and 33** the *Bonola* reference discloses a method of emulating a device (*peripheral*) loading an emulation driver for the device, and dynamically exposing the emulated device functionality (**Col. 1 Lines 65-67, Col. 2 Lines 1-31**).

However, the *Bonola* reference does not expressly disclose, a **Virtual Device Object** and emulating a device by a node in a serial bus.

An artisan of ordinary skill would have been motivated to search the related device driver art to find a object oriented methods of abstracting computer peripheral devices in software for the purpose of developing device drivers before that actual hardware was available (*Bonola Col. 1 lines 40-63*). In the same art of emulating hardware devices in software, the *Tushie et al.* reference teaches a **Virtual Device Object (Figure 2 & 4, Col. 3 Lines 10-23)**.

Thus, it would have been obvious, to one of ordinary skill in the art, at the time the invention was made; to have combined the device emulation technologies of the *Bonola* reference with the object oriented object abstraction technologies of the *Tushie et al.* reference because, by allowing for easy configuration change to take place on a virtual model of a device rather than the device itself, changes can be made without having to use a complex programming language which would add complexity and expense (**Col. 3 Lines 19-24, Col. 4 Lines 45-55**).

An artisan of ordinary skill would have been motivated to write device drivers for providing a software abstraction layer to devices on a node of a serial bus to support the newer IEEE1394 technologies (*Bonola Col. 1 lines 40-63*). In the technology are of configuration of peripherals on a bus, the *Staats* reference discloses device drivers providing support for devices on nodes on a serial bus (*Staats, Figures 1, 4, 7, Col. 2 Lines 44-52*).

Thus, it would have been obvious, to one of ordinary skill in the art, at the time the invention was made, to have combined the device emulation technologies of the *Bonola* reference with the control of nodes on a serial bus technologies of the *Staats* reference because, by providing an improved method of using key space a large number of peripherals can be supported on a IEEE 1394 bus (**Col. 2 Lines 31-36**).

5.2 In regards to dependent **Claim 35** and the limitation of issuing a bus reset in independent **Claim 17** the *IEEE 1394* standard for a high speed serial bus teaches issuing bus resets and “*OFFICIAL NOTICE*” bus resets are known in the art as methods of initializing a plurality of devices on the bus and enumerating them. (*The Examiner notes that the IEEE 1394 bus standard is a 394-page document and in the interest of compact prosecution will not provide the document in this office action but will be happy to provide the document if the Applicant requests a copy.*)

5.3 As regards dependent **Claims 2 and 34**, the *Bonola* reference does not expressly disclose enumeration by the node of a serial bus.

The *Staats* reference discloses enumeration by the node of a serial bus (**Figures 1, 7, 8, 8A, 9 & 10, Col. 2 Lines 44-53, Col. 4 Lines 47-53**), it is noted by the Examiner that hierarchical fashion discussed in the *Staats* prior art reference is a form of enumeration.

5.4 As regards dependent **Claim 3** the *Bonola* reference does not expressly disclose a physical device object.

The *Tushie et al.* reference discloses a physical device object (**Figure 2 Item 203**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made to have modified the device emulation technologies of the *Bonola* reference with the object oriented programming technologies of the *Tushie et al.* reference because by abstracting the devices on the serial bus using software objects the many complex details of interacting with each device can be abstracted so that any programmer writing software to interact with a particular device doesn't have to know all of the details of what is involved in operating the device when that programmer needs to focus on only that functionality that is required to perform the programming task required.

5.5 As regards dependent **Claim 6** the *Bonola* reference does not expressly disclose a device being plugged into a serial bus.

The *Staats* reference discloses a device being plugged into a serial bus (**Figure 4, Col. 1 Lines 15-27, Col. 4 Lines 7-21**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made, to have modified the device emulation methods of the *Bonola* reference with the serial bus technologies of the *Staats* reference because, a artisan of ordinary skill would have known of the 1394 bus methods of "hot" plugging new serial devices into the bus and the artisan would have relied on the methodologies of the *Bonola* reference in order to implement support for hot plugging serial devices into a 1394 serial bus after the host computer system haad already completed the "boot" process.

5.6 As regards dependent **Claims 7 and 36** the *Bonola* reference does not expressly disclose the 1394 bus standard.

The *Staats* reference discloses the 1394 bus standard (**Col. 1 Lines 15-27**).

In regards to motivation to combine, please refer to paragraph 2.4 above.

5.7 As regards dependent **Claims 8 and 9** the *Bonola* reference discloses the *virtualization* of a peripheral device that is not affected by any bus events including addition or removal of a device (**Figure 10, Col. 17 Lines 19-48**). *Specifically*, the Examiner notes that the *Bonola* reference is teaching an emulated *virtual* device that is not affected by the functioning of other *real* devices and in fact, as disclosed in the cited lines and columns above, performs more efficiently because of the use of separate CPU's.

5.8 As regards dependent **Claims 10 and 11** the *Bonola* reference does not expressly disclose the node is a general-purpose computer running a general-purpose operating system or the emulated device functionality is done by configuration memory.

The *Staats* reference discloses the node is a general-purpose computer running a general-purpose operating system (**Figure 4, Col. 2 Lines 39-43, Col. 6 Lines 31-45**), or the emulated device functionality is done by configuration memory (**Figure 7 Col. 6 Lines 46-56**).

5.9 As regards dependent **Claim 18** the *Bonola* reference does not expressly disclose adding a unit directory to the configuration memory.

The *Staats* reference discloses adding a unit directory to the configuration memory (**Figures 1, 7, Col. 2 Lines 44-53**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made to have combined the *device emulation* methods of the *Bonola* reference with the *unit directory configuration* methods of the *Staats* reference because of the need to preserve "*key space*" (**Staats Col. 2 Lines 31-36**).

Art Unit: 2123

6. Dependent **Claims 4 and 5** are rejected under **35 U.S.C. 103(a)** as being unpatentable over **Bonola U.S. Patent 5,717,903** in view of **Tushie et al. U.S. Patent 6,202,155** and in further view of **Staats U.S. Patent 5,968,152** and in further view of **Carter et al. U.S. Patent 5,996,050**.

6.1 As regards independent **Claim 1**, see paragraph 5.1 above.

6.2 As regards dependent **Claim 4** the *Bonola* reference does not expressly disclose bus objects that then create other device objects.

The *Carter et al.* reference discloses bus objects creating other device objects (**Figure 1B, items 204 and 600, note the pointers to items 218 and 220, also note the words "Bus Cycle Object Creation"**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made, to have combined the peripheral emulation methods of the *Bonola* reference with the computer bus abstraction methods of the *Carter et al.* reference because, object oriented design allow for reliable and efficient methods of monitoring system buses (***Carter et al. Col. 3 Lines 53-55***).

6.3 As regards dependent **Claim 5** the *Bonola* reference does not expressly disclose a 1394 bus object.

The *Staats* reference discloses a 1394 bus (**Figure 4**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made, to have modified the peripheral emulation methods of the *Bonola* reference with the 1394 bus methods of the *Staats* reference because, object abstraction methods are known by ordinary artisans and these methods provide advantages in abstracting underlying

Art Unit: 2123

software functionality that decreases the amount of time artisans are required to code software, so that new technologies, like “*Firewire*” or IEEE 1394 bus technologies can be quickly reduced to practice and implemented into today’s computing products.

7. Independent **Claims 19 and 27** and dependent **Claims 20, 21, 26, 29 and 32** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Staats U.S. Patent 5,968,152** in view of **Bonola U.S. Patent 5,717,903**.

7.1 In regards to independent **Claims 19 and 27** the *Staats* reference teaches a serial bus (**Figure 4, Col. 1 Lines 15-28**), and a node connected to the serial bus (**Figure 4 Items 74, 78 82 or 80**) and in regards to the limitation of teaching a configuration memory compliant with the IEEE-1212 standard (**Col. 1 Lines 35-46**).

However the *Staats* reference does not expressly disclose emulation of a device on the serial bus.

The *Staats* reference discloses that there are problems with the number of future devices that can be added, do to limited key space (**Col. 4 Lines 61-64**), and that a driver image can be loaded from media that is plugged into the bus (**Col. 6 Lines 46-56**). An artisan of ordinary skill, knowing the *IEEE 1394* standard bus architecture accommodates hot removal and insertion of devices would want a method of providing a “*place holder*” for a device that has been temporarily removed but which can reasonably be assumed will be re-inserted at a future time. The artisan would not want the user of the computer system to have to wait while the device drive for the device at that particular node would have to be loaded and unloaded every time that particular peripheral was inserted into the bus. Thus, an artisan would look for a mechanism to

Art Unit: 2123

represent that device on the serial bus with out the actual device actually being present. In the related art of peripheral emulation the *Bonola* reference teaches the emulation of a peripheral device that is not actually present (**Col. 1 Lines 65-67, Col. 2 Lines 1-31**).

Thus, it would have been obvious, to one of ordinary skill in the art, at the time the invention was made, to have combined the serial bus methods of the *Staats* reference with the peripheral emulation method of the *Bonola* reference because, by emulating a peripheral on the 1394 bus, the device driver for that device doesn't have to be reloaded every time the device is inserted into the bus, thus allowing the computer system to operate more efficiently.

7.2 As regards dependent **Claims 20 and 29** the *Staats* reference discloses an IEEE 1394 bus (**Col. 1 Lines 15-27**).

7.3 As regards dependent **Claim 21** the *Staats* reference discloses an IEEE 1212 compliant memory device (**Col. 1 Lines 35-46**).

7.4 As regards dependent **Claim 26** the *Staats* reference discloses a physical device (**Figure 4**).

7.5 As regards dependent **Claim 32** the *Staats* reference discloses a directory for a device (**Figure 1, Item 20**).

8. Dependent **Claims 22 and 23** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Staats U.S. Patent 5,968,152** in view of **Bonola U.S. Patent 5,717,903** and in further view of **Carter et al. U.S. Patent 5,996,050**.

8.1 As regards independent **Claim 19** see paragraph 7.1 above.

8.2 As regards dependent **Claim 21** see paragraph 7.3 above.

8.3 As regards dependent **Claim 22** the *Staats* reference does not expressly disclose a bus driver and device objects used for communications on the bus.

The *Carter et al.* reference discloses a bus driver and device objects used for communications on the bus (**Figure 1B**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made; to have combined the serial bus teachings of the *Staats* reference with the object oriented bus control methods of the *Carter et al.* reference because, object oriented methods of abstracting hardware devices provides a simple and efficient method of sending and receiving data to and from hardware devices without the programmer having to know all of the details of how the hardware device functions.

8.4 As regards dependent **Claim 23** the *Staats* reference discloses a 1394 based bus (**Col. 1 Lines 15-26**).

9. Dependent **Claims 24 and 25** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Staats U.S. Patent 5,968,152** in view of **Bonola U.S. Patent 5,717,903** and in further view of **Carter et al. U.S. Patent 5,996,050** and in further view of **Tushie et al. U.S. Patent 6,202,155**.

9.1 As regards independent **Claim 19** see paragraph 7.1 above.

9.2 As regards dependent **Claim 21** see paragraph 7.3 above.

9.3 As regards dependent **Claim 22** see paragraph 8.3 above.

9.4 As regards dependent **Claims 24 and 25**, the *Staats* reference does not expressly disclose virtual device objects and virtual device drivers.

The *Tushie et al.* reference discloses virtual device objects and virtual device drivers (**Figure 2, Col. 3 Lines 10-24**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made, to have modified the serial bus methods of the *Staats* reference with the device virtualization methods of the *Tushie et al.* reference because, the methods in the *Tushie et al.* reference are useful for abstracting computing devices connected both locally and remotely (*Tushie et al. Col. 4 Lines 45-55*).

10. Dependent **Claims 28** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Staats U.S. Patent 5,968,152** in view of **Bonola U.S. Patent 5,717,903** and in further view of **Carter et al. U.S. Patent 5,996,050**.

10.1 In regards to independent **Claim 27** see paragraph 7.1 above.

10.2 In regards to dependent **Claim 28** the *Staats* reference does not expressly disclose bus driver and device objects used for communications on a bus.

The *Carter et al.* reference discloses a bus driver and device objects used for communications on the bus (**Figure 1B**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made; to have combined the serial bus teachings of the *Staats* reference with the object oriented bus control methods of the *Carter et al.* reference because, object oriented methods of abstracting hardware devices provides a simple and efficient method of sending and receiving data to and from hardware devices without the programmer having to know all of the details of how the hardware device functions.

11. Dependent **Claims 30 and 31** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Staats U.S. Patent 5,968,152** in view of **Bonola U.S. Patent 5,717,903** and in further view of **Tushie et al. U.S. Patent 6,202,155**.

11.1 As regards independent **Claim 27** see paragraph 7.1 above.

11.2 As regards dependent **Claims 30 and 31**, the *Staats* reference does not expressly disclose virtual device objects and virtual device drivers.

The *Tushie et al.* reference discloses virtual device objects and virtual device drivers (**Figure 2, Col. 3 Lines 10-24**).

It would have been obvious, to one of ordinary skill in the art, at the time the invention was made, to have modified the serial bus methods of the *Staats* reference with the device virtualization methods of the *Tushie et al.* reference because, the methods in the *Tushie et al.* reference are useful for abstracting computing devices connected both locally and remotely (***Tushie et al. Col. 4 Lines 45-55***).

Allowable Subject Matter

12. Dependent **Claims 37, 38 and 39** are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

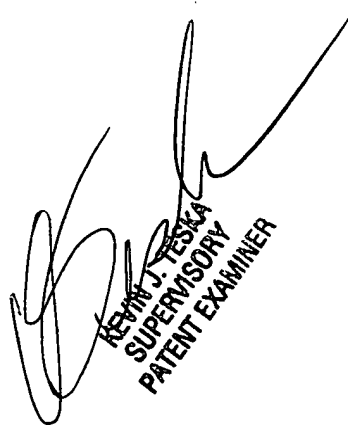
13. Claims 1-11, 17-36 are rejected. Claims 12-16 are cancelled. Claims 37, 38 and 39 are objected to. This action is **NON-Final**.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dwin M Craig whose telephone number is (571) 272-3710. The examiner can normally be reached on 10:00 - 6:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin Teska can be reached on (571) 272-3716. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DMC



KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER